Appl. No. 09/851,940   PATENT
Amdt. dated October 5, 2004
Reply to Office Action of May 5, 2004

Amendments to the Claims:

Claims 1, 10, 13, and 22 have been amended. Claims 2-3, 7-9, 11-12, 14-15, 19-21, 23-24, 27-30 have been canceled without prejudice. This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1   1. (Currently Amended) A method of analyzing intersections between objects in
2   computer animation comprising:
3   providing a plurality of objects represented by a plurality of meshes, with each of
4   said plurality of objects being represented by one of said plurality of meshes and each of said
5   meshes being formed by a set of vertices, where a set of pairs of vertices of said set of vertices
6   define a set of edges;
7   checking all edges of said meshes to determine if said set of edges of said meshes
8   intersect with any of said plurality of meshes;
9   tracing an intersection path formed by intersection of said edges with any of said
10 plurality of meshes;
11   determining which vertices of said meshes are contained within said intersection
12 path, the determining comprising:
13   selecting an arbitrary edge of a mesh that crosses said intersection path,
14 where said arbitrary edge is formed by vertices u and v and where said mesh contains said
15 intersection path;
16   performing a search of said mesh, radiating from one of said vertices u and
17 v, identifying all vertices in all of said edges that cross said intersection path, and defining the set
18 of vertices identified as a playpen;
19   tracing said intersection path and identifying vertices, within said playpen,
20 on a left side of said intersection path as left and retracing said intersection path in an opposite
21 direction and identifying vertices, with said playpen, on a right side of said intersection path as
22 right;

Appl. No. 09/851,940
Amdt. dated October 5, 2004
Reply to Office Action of May 5, 2004

PATENT

23   determining whether vertices adjacent to vertices identified as left and

24 right lie outside of said playpen;

25   discarding said intersection path when at least one of both left and right

26 identified vertices lie outside said playpen; and

27   determining whether at least one vertex adjacent to said right identified

28 vertices lies outside said playpen; and

29   characterizing each of said left identified vertices as contained within said

30 intersection path when at least one vertex adjacent to said right identified vertices lies outside

31 said playpen and characterizing each of said right identified vertices as contained within said

32 intersection path when at least one vertex adjacent to said left identified vertices lies outside said

33 playpen; and

34   setting a polarity of each vertex contained within said intersection path to indicate

35 that said vertex is contained within said intersection path, wherein a polarity of a vertex is set

36 based upon a number of disconnected regions formed by said intersection path.

1   2. (Canceled)

1   3. (Canceled)

1   4. (Previously Presented) The method of claim 1 wherein said intersection path is

2 a self-intersection with the intersection path being contained in a single mesh and wherein setting

3 a polarity of each vertex comprises:

4   associating a first color with each vertex that is contained within said intersection

5 path when the intersection yields one region; and

6   when the intersection yields first and second disconnected regions, associating a

7 second color with each vertex contained in said first disconnected region and associating a third

8 color with each vertex contained in the second disconnected region.

1   5. (Previously Presented) The method of claim 1 wherein said intersection path is

2 an intersection between a first mesh and a second mesh and wherein setting a polarity of each

3   vertex comprises associating a first color with vertices of the first mesh contained within said

4   intersection path and associating a second color with vertices of the second mesh contained

5   within said intersection path.

1   6. (Previously Presented) The method of claim 4, further comprising displaying

2   said objects on a computer display with vertices displayed in colors associated with the vertices.

1   7. (Canceled)

1   8. (Canceled)

1   9. (Canceled)

1   10. (Currently Amended) A method of simulating motion of objects in computer

2   animation, the method comprising:

3   providing a plurality of objects represented by a plurality of meshes, with each of

4   said plurality of objects being represented by one of said plurality of meshes and each of said

5   meshes being formed by a set of vertices, where at least one of said objects is an animated object

6   and at least one of said objects is a simulated object;

7   positioning said objects at some time *t* to provide one frame of said computer

8   animation;

9   determining an intersection path formed by intersections between said objects;

10   setting a polarity of each vertex contained within said intersection path based

11   upon a number of disconnected regions formed by said intersection path, wherein setting the

12   polarity comprises:

13   when said intersection path is a self-intersection with the intersection path

14   contained in a single mesh,

15   associating a first color with each vertex contained within said

16   intersection path when the intersection yields one region, and

17   when the intersection yields first and second disconnect regions,

18   associating a second color with each vertex contained in said first disconnected region and a

19   third color with each vertex contained in the second disconnected region; and

20   when said intersection path is an intersection between a first mesh and a

21   second mesh, associating the second color with each vertex of the first mesh contained within

22   said intersection path and associating a third color with each vertex of the second mesh contained

23   within said intersection path;

24   setting a simulated force between vertices of said at least one simulated object

25   based on the polarity set for said vertices of said at least one simulated object, wherein setting the

26   simulated force comprises:

27   setting said simulated force to cause an attraction between vertices of said

28   at least one simulated object when said vertices are associated with said second or third colors;

29   setting said simulated force to cause a repulsion between vertices of said at

30   least one simulated object when said vertices are not associated with said first color, said second

31   color, or said third color; and

32   setting said simulated force to cause neither an attraction or a repulsion

33   between vertices of said at least one simulated object when said vertices are associated with said

34   first color; and

35   advancing the computer animation to a time $t + \Delta t$ and simulating motions of said

36   objects using said simulated force to simulate motions of said at least one simulated object.

1   11. (Canceled)

1   12. (Canceled)

1   13. (Currently Amended) A computer program product stored on a computer

2   readable medium for processing computer generated objects, said computer program product

3   comprising:

4   code for providing a plurality of objects represented by a plurality of meshes, with

5   each of said plurality of objects being represented by one of said plurality of meshes and each of

Appl. No. 09/851,940   PATENT
Amdt. dated October 5, 2004
Reply to Office Action of May 5, 2004

6   said meshes being formed by a set of vertices, where a set of pairs of vertices of said set of

7   vertices define a set of edges;

8   code for checking all edges of said meshes to determine if said set of edges of said

9   meshes intersect with any of said plurality of meshes;

10   code for tracing an intersection path formed by intersection of said edges with any

11   of said plurality of meshes; and   .

12   code for determining which vertices of said meshes are contained within said

13   intersection path, the code for determining comprising:

14   code for selecting an arbitrary edge of a mesh that crosses said intersection

15   path, where said arbitrary edge is formed by vertices u and v and where said mesh contains said

16   intersection path;

17   code for performing a search of said mesh, radiating from one of said

18   vertices u and v, identifying all vertices in all of said edges that cross said intersection path, and

19   defining the set of vertices identified as a playpen;

20   code for tracing said intersection path and identifying vertices, within said

21   playpen, on a left side of said intersection path as left and retracing said intersection path in an

22   opposite direction and identifying vertices, with said playpen, on a right side of said intersection

23   path as right;

24   code for determining whether vertices adjacent to vertices identified as left

25   and right lie outside of said playpen;

26   code for discarding said intersection path when at least one of both left

27   and right identified vertices lie outside said playpen;

28   code for determining whether at least one vertex adjacent to said right

29   identified vertices lies outside said playpen; and

30   code for characterizing each of said left identified vertices as contained

31   within said intersection path when at least one vertex adjacent to said right identified vertices lies

32   outside said playpen and characterizing each of said right identified vertices as contained within

33   said intersection path when at least one vertex adjacent to said left identified vertices lies outside

34   said playpen; and

35   code for setting a polarity of each vertex contained within said intersection path to

36   indicate that said vertex is contained within said intersection path, wherein a polarity of a vertex

37   is set based upon a number of disconnected regions formed by said intersection path.

1   14. (Canceled)

1   15. (Canceled)

1   16. (Previously Presented) The computer program product of claim 13 wherein

2   said intersection path is a self-intersection with the intersection path being contained in a single

3   mesh and the code for setting a polarity of each vertex comprises:

4   code for associating a first color with each vertex that is contained within said

5   intersection path when the intersection yields one region; and

6   when the intersection yields first and second disconnected regions, code for

7   associating a second color with each vertex contained in said first disconnected region and

8   associating a third color with each vertex contained in the second disconnected region.

1   17. (Previously Presented) The computer program product of claim 13 wherein

2   said intersection path is an intersection between a first mesh and a second mesh and the code for

3   setting a polarity of each vertex comprises code for associating a first color with vertices of the

4   first mesh contained within said intersection path and code for associating a second color with

5   vertices of the second mesh contained within said intersection path.

1   18. (Previously Presented) The computer program product of claim 16, further

2   comprising code for displaying said objects on a computer display with vertices displayed in

3   colors associated with the vertices.

1   19. (Canceled)

1   20. (Canceled)

1   21. (Canceled)

Appl. No. 09/851,940
Amdt. dated October 5, 2004
Reply to Office Action of May 5, 2004

PATENT

1   22. (Currently Amended) A computer program product stored on a computer

2   readable medium for processing computer generated objects in computer animation, said

3   computer program product comprising:

4   code for providing a plurality of objects represented by a plurality of meshes, with

5   each of said plurality of objects being represented by one of said plurality of meshes and each of

6   said meshes being formed by a set of vertices, where at least one of said objects is an animated

7   object and at least one of said objects is a simulated object;

8   code for positioning said objects at some time t to provide one frame of said

9   computer animation;

10   code for determining an intersection path formed by intersections between said

11   objects;

12   code for setting a polarity of each vertex contained within said intersection path

13   based upon a number of disconnected regions formed by said intersection path, wherein the code

14   for setting a polarity of each vertex comprises:

15   when said intersection path is a self-intersection with the intersection path

16   contained in a single mesh,

17   code for associating a first color with each vertex contained within

18   said intersection path when the intersection yields one region and

19   when said intersection path yields first and second disconnected

20   regions, code for associating a second color with each vertex contained in said first disconnected

21   region and a third color with each vertex contained in the second disconnected region; and

22   when said intersection path is formed from an intersection between a first

23   mesh and a second mesh, the code for setting a polarity of each vertex contained in said

24   intersection path comprises code for associating the second color with each vertex of the first

25   mesh and associating the third color with each vertex of the second mesh;

26   code for setting a simulated force between vertices of said at least one simulated

27   object based on the polarity set for said vertices of said at least one simulated object, wherein the

28   code for setting a simulated force comprises:

PATENT

29   code for setting said simulated force to cause an attraction between

30   vertices of said at least one simulated object when said vertices are associated with said second

31   or third colors;

32   code for setting said simulated force to cause a repulsion between vertices

33   of said at least one simulated object when said vertices are not associated with said first color,

34   said second color, or said third color; and

35   code for setting said simulated force to cause neither an attraction or a

36   repulsion between vertices of said at least one simulated object when said vertices are associated

37   with said first color; and

38   code for advancing the computer animation to a time $t + \Delta t$ and simulating

39   motions of said objects using said simulated force to simulate motions of said at least one

40   simulated object.

1   23. (Canceled)

1   24. (Canceled)

1   25.   (Previously Presented) The method of claim 5, further comprising

2   displaying said objects on a computer display with vertices displayed in colors associated with

3   the vertices.

1   26.   (Previously Presented) The computer program product of claim 17, further

2   comprising code for displaying said objects on a computer display with vertices displayed in

3   colors associated with the vertices.

1   27.   (Canceled)

1   28.   (Canceled)

1   29.   (Canceled)

1   30.   (Canceled)